# A Causal Rhythm Grouping

Jensen, Karl Kristoffer

*Published in:*
Lecture Notes in Computer Science

*Publication date:*
2005

*Document Version*
Early version, also known as pre-print

# A Causal Rhythm Grouping

Kristoffer Jensen

Department of Computer Science, University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen, Denmark
krist@diku.dk

**Abstract.** This paper presents a method to identify segment boundaries in music. The method is based on a hierarchical model; first a features is measured from the audio, then a measure of rhythm is calculated from the feature (the rhythmogram), the diagonal of a self-similarity matrix is calculated from the rhythmogram, and finally the segment boundaries are found on a smoothed novelty measure, calculated from the diagonal of the self-similarity matrix. All the steps of the model have been accompanied with an informal evaluation, and the final system is tested on a variety of rhythmic songs with good results. The paper introduces a new feature that is shown to work significantly better than previously used features, a robust rhythm model and a robust, relatively cheap method to identify structure from the novelty measure.

## 1 Introduction

As more and more of the music delivery and playback are made through computers, it has become necessary to introduce computer tools for the common music tasks. This includes, for instance, common tasks such as music playback, control and summary. This paper presents a novel approach to the music segmentation tasks based on rhythm modeling. Music segmentation is here seen as the identification of boundaries between common segments in rhythmic music, such as intro, chorus, verse, etc. These boundaries often consist of changes in the rhythm. The segmentation work undertaken here introduces structure in the music, whereas the previous work [15], on which this work is partly based, mainly investigated the tempo.

The segmentation is useful for many tasks. This approach, which is both real-time and not too processor intensive, is useful in real-time situations. One use is to perform live recomposition, using for instance Pattern Play [20], where the found segments is reintroduced into the music, potentially after some effects performed on the segment. Another use is to assists Djs in computer based DJ software, such as Mixxx [1], for beat mixing, intro skipping, or other uses.

The current approach is built on previous work in beat and tempo estimation [15], where a Beat Histogram was used to estimate the tempo. Only the maximum of the beat histogram was used. In this work, the full histogram is calculated for each time frame. The self-similarity [8, 9] of the histogram, which is here called a rhythmogram, is calculated, and a measure of novelty [9] is extracted. The novelty measure is only calculated on the diagonal of the self-similarity matrix, which thus necessitates only the calculation of a small subset of the full matrix. Finally the segments are found by smoothing the novelty measure, identifying the peaks (the segment boundaries), and following them to the unsmoothed case in several steps using a technique borrowed from edge detection in image scale-space.

Several authors have presented segmentation and visualization of music using a self-similarity matrix [10, 2, 21] with good results. Other methods to segment music include information-theoretic methods [7], or methods inspired from ICA [3].

When designing a music section grouping, or section-clustering algorithm, it is intuitive to try to understand what knowledge there is about how humans go about doing the same task. Desain [6] introduced the decomposable theory of rhythm, in which rhythm is perceived by all note onsets, in what he modeled as essentially an autocorrelation step. Scheirer [23] made some *analysis by synthesis* experiments, and determined that rhythm could not be perceived by amplitude alone, but needed some frequency dependent information, which he constructed using six band-pass filters. No experiments were done using filtered signals, by varying only the filter cutoff frequency. This would make probably the success of one amplitude-based feature, if it were suitably weighted by e.g. an equal loudness contour, or the spectral centroid, which weights higher frequencies higher. Several studies have investigated the influence of timbre on structure. [19] found that timbre did not affect the recognition of familiar melodies, but that it did hurt recognition on non-familiar melodies. McAdams [18], studied contemporary and tonal music, and found that the orchestration affects the perceived similarity of musical segments strongly in some cases. He also found that musically trained listeners find structure through surface features (linked to the instrumentation) whereas untrained listeners focused on more abstract features (melodic contour, rhythm). This helped non-musicians recognize music with a modified timbre (piano and chamber music versions). Deliège and Mélen [5] postulates that music is segmented into sections of varying length using cue abstraction mechanism, and the principle of sameness and difference, and that the organization of the segmentation, reiterated at different hierarchical levels, permits the structure to be grasped. The cues (essentially motifs in classical music, and acoustic, instrumental, or temporal otherwise) act as reference points during long time spans. Deliège and Mélen furthermore illustrate this cue abstraction process through several experiments, finding, among other things, that musicians are more sensitive to structural functions, and that the structuring process is used for remembering, in particular, the first and last segment.

Desain thus inspired the use of an autocorrelation function for the rhythm modeling; Scheirer showed the necessity to model the acoustic signal somehow akin to human perception. For simplicity and processing reasons a scalar feature, which does indeed perform satisfactory, is used in this work Deliège and Mélen inspired the use of a hierarchical model presented here, consisting of a feature, calculated from the acoustic signal, a time varying rhythm abstraction, a self-similarity matrix, and a novelty function extracted from the self-similarity matrix.

This paper is organized in the following manner. Section two presents the beat estimation work that is used to find the optimal feature, and introduces the measure of rhythm, section three presents the self-similarity applied to the rhythm, section four gives an overview of the rhythm grouping in one song. In section 5, an evaluation is performed, and finally there is a conclusion.

## 2 A measure of rhythm

Rhythm estimation is the process of determining the musical rhythm from a representation of music, symbolic or acoustic. The problem of automatically finding the rhythm includes, as a first step, finding the onsets of the notes. This approach is used here to investigate the quality of the audio features. The feature that performs best is furthermore used in the rhythm model.

### 2.1 Beat and tempo

The beat in music is often marked by transient sounds, e.g. note onsets of drums or other instrumental sounds. Onset positions may correspond to the position of a beat, while some onsets fall off beat. The onset detection is made using a feature estimated from the audio, which can

subsequently be used for the segmentation task. In a previous work [15], the high frequency content was found to perform best, and was used to create a beat histogram to evaluate the beat. Other related works include Goto and Muraoka [11] who presented a beat tracking system, where two features were extracted from the audio based on the frequency band of the snare and bass drum. Later Goto and Muraoka [12] developed a system to perform beat tracking independent of drum sounds, based on detection of chord changes. Scheirer [23] took another approach, by using a non-linear operation of the estimated energy of six band-pass filters as features. The result was combined in a discrete frequency analysis to find the underlying beat. As opposed to the approaches described so far Dixon [7] build a non-causal system, where an amplitude based feature was used as clustering of inter-onset intervals. By evaluating the inter-onset intervals, hypothesis is formed and one is selected as the beat interval. This system also gives successful results on simpler musical structures. Laroche [14] built an offline system, using one features, the energy flux, cross-correlation and dynamic programming, to estimate the time-varying tempo.

### 2.2 Feature Comparison

There have been a large number of possible features proposed for the tasks and tempo estimation and segmentation. This section introduces a new scalar feature, the Perceptual Spectral Flux, and show that it performs better in note-onset detection than other features.

Apart from the possible vector sets (Chroma, MFCC, PLP, etc), [15] evaluated a number of different scalar features for use in beat estimation systems. The approach was to identifying a large number of audio features, and subsequently evaluating the quality of the features using error measures. A number of music pieces were manually marked, by identifying the note transients, and these marks were used when evaluating the features. In [15], the high frequency content (HFC) [17] was found to perform best. In this work, however, another feature has been evaluated, which performs better than the HFC. This feature, here called the perceptual spectral flux (PSF), is calculated as

$$PSF_n = \sum_{k=1}^{N_b/2} W_B \left( \left( a_k^n \right)^{1/3} - \left( a_k^{n-1} \right)^{1/3} \right), \tag{1}$$

where $n$ is the block index, and $N_b$ is the block size, and $a_k$ is the magnitude of the Short-Time Fourier Transform (STFT), obtained using a hanning window. $W_b$ is the frequency weighting used to obtain a value closer to the human loudness contour, and the power function is used to simulate the intensity-loudness power law. The power function furthermore reduces the random amplitude variations. These two steps are inspired from the PLP front-end [13] used in speech recognition.

The error measures used in the evaluation is the signal to noise ratio (S/N), calculated as the ratio between the sum of the *hills* (corresponding to the peaks and corresponding slopes) of the peaks of the feature under test that are matched to a manual mark to the sum of those that are not, and the matched ratio, calculated as the number of matched peaks, divided by the number of manual marks. The feature peaks are chosen as all local maximums above a given running threshold. As the threshold is increased, the signal to noise increased, whereas the matched ratio decreases. The thresholds necessary to obtain an arbitrary value of 75 % matched peaks (which is possible in almost all cases) are found for all features, and the signal to noise ratio is compared for this threshold. In [15], the high frequency content (HFC) was found to have twice as good S/N ratio as the other measured features. Using the same material, the PSF performs twice as good as the HFC. This can be tentatively explained as, since the HFC weight the high frequency most, it indicates mainly the hihat, and the transient instruments, such as the piano. The spectral flow, with no

frequency weighting, essentially favors the low frequencies, since these generally have significantly more energy than the mid, or high frequencies. The PSF weight everything approximately as the human ear, and would then indicate both the high frequency sounds, but also the low frequency sounds, such as the bass, or other instrumental sounds with less transient behavior.

The PSF is calculated on a block of 20 msec., with a step size of 10 msec. An example of the PSF, calculated on an excerpt of Train to Barcelona[1], can be seen in figure 1.

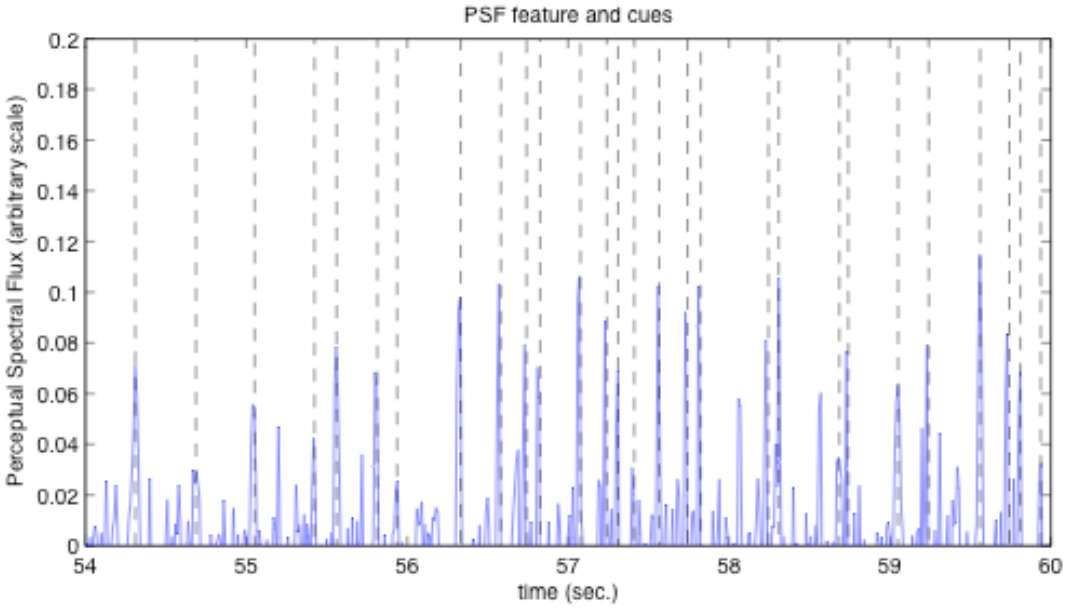


**Figure 1. Example of PSF feature, and manually marked note onset marks (dashed vertical lines) for the piece *Train to Barcelona*.**

### 2.3 Rhythmogram

The PSF feature indicates most of the manual marks correctly, but it has many peaks that does not corresponds to note onset, and many note onset does not have a peak in the PSF. In order to get a more robust rhythm feature, the autocorrelation of the feature is now calculated on overlapping blocks of 8 seconds, with half a second overlap. Only the information between zero and two seconds is retained. The autocorrelation is normalized so that autocorrelation at zero lag equals one. This effectively prevents loudness variations to have any influence. Other presented models of rhythm include [21], which uses an FFT on the energy output of the auditory filterbanks, and [22], whose rhythm patterns consist of the FFT coefficients of the critical band outputs. The autocorrelation has been chosen, instead of the FFT used by the two above-mentioned papers, for two reasons, first, it is believed to be used in the human perception of rhythm [6], and second, it is believed to be more easily understood visually.

[1] By Akufen. Appearing on Various - Elektronische Musik - Interkontinental (Traum CD07), December 2001

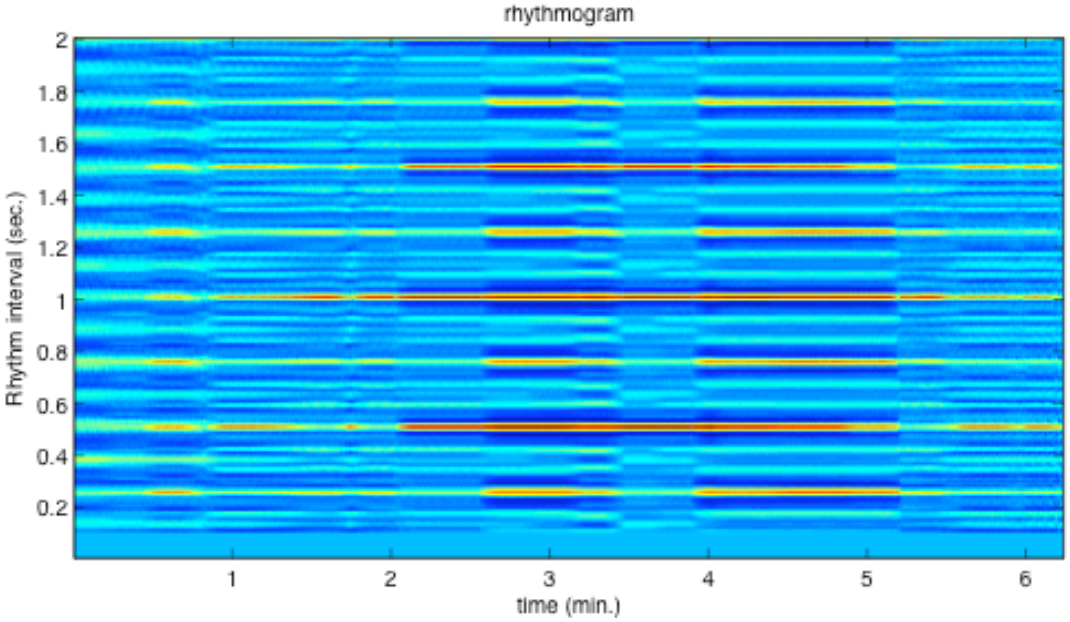


**Figure 2. Rhythmogram for *Train to Barcelona*.**

If visualized with lag time on the y-axis, time position on the x-axis, and the autocorrelation values visualized as colors, it gives a fast overview of the rhythmic evolution of a song. This representation, here called a rhythmogram, can give much information about the rhythm and the evolution of the rhythm in time. An example of the rhythmogram for *Train to Barcelona* is shown in figure 2. The song seems to be a 4/4 with a tempo of 240 BPM, but in practice, the perceived beat is 120 BPM. In the first minute, it has an additional 8$^{th}$ beat, which is transformed into a 12$^{th}$ beat for the rest of the song, except a short period between 3 1/2 and 4 minutes, approximately.

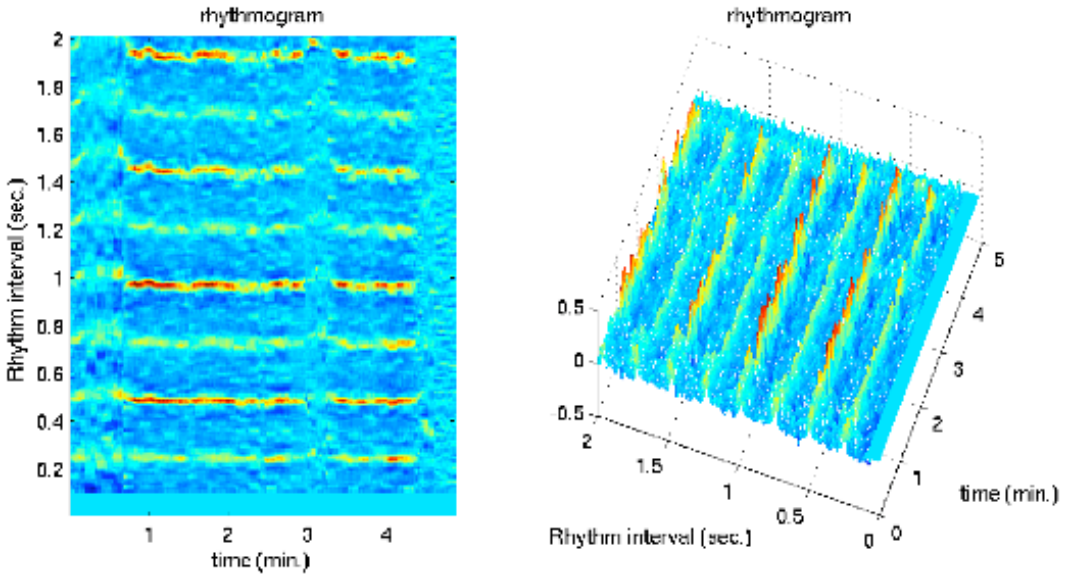


**Figure 3. 2D and 3D rhythmogram for *I must be dreaming*.**

Although the rhythmogram seems like a stable and robust representation, it can easily be shown that the robustness is, in part, caused by the gestalt behavior of the visual system. Indeed, if seen from another angle (in a 3D visualization), the rhythmogram reveals more movement, i.e. changes in relative strength of each beat in the measure, thus sometimes having different predominant beats in the measure. An example of such a 3D plot for I must be Dreaming, by Mink de Ville is shown in figure 3 (right). It is clear that it is not easy to segment the song according to a difference in rhythm. There seem to be an intro the first half minute, possibly repeated at around 3 minutes. Some change is taking place at around 1 1/2, 2 1/2 and 4 minutes, each time followed by a small change in tempo. As the song seemed to be played live, there is inherently an uncertainty in tempo, rhythm strength of each beat, and other timbre phenomena, which is all influencing to some degree on the rhythmogram.

## 3 Selfsimilarity

In order to get a better representation of the similarity of the song segments, a measure of self-similarity is used.

Several studies have used a measure of self-similarity [8] in automatic music analysis. Foote [10] used the dot product on MFCC sampled at a 100 Hz rate to visualize the self-similarity of different music excerpt. Later he introduced a checkerboard kernel correlation as a novelty measure [9] that identifies notes with small time lag, and structure with larger lags with good success. Bartsch and Wakefield [2] used the chroma-based representation (all FFT bins are put into one of 12 chromas) to calculate the cross-correlation and identify repeated segments, corresponding to the chorus, for audio thumbnailing. Peeters [21] calculated the self-similarity from the FFT on the energy output of an auditory filterbank.

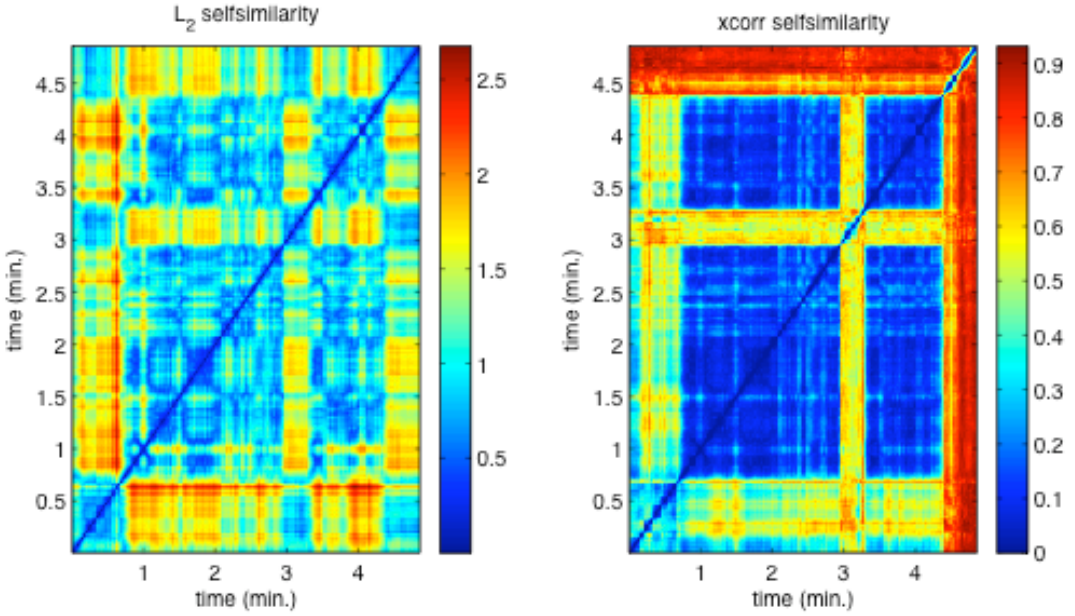


**Figure 4. $L_2$ norm (left) and cross-correlation (right) self-similarity for *I must be dreaming*.**

Generally, this measure of self-similarity is calculated directly on the feature(s), but in this case, an extra parameterization is introduced, the rhythmogram. The low sampling rate of the

rhythmogram permits to calculate a rather small self-similarity matrix that is faster to calculate and easier to manipulate. In addition, as the rhythmogram seems to be close to a human perception of rhythm (cf. Desains decomposable theory of rhythm [6]), this intermediate step is also believed to make the self-similarity more directed towards rhythm than other features of the song, such as timbre. As the self-similarity should work, even if there is a drift in tempo, the cross-correlation self-similarity method is used, albeit it is significantly slower than the $L_2$ norm method. This has also been shown to minimize the $L_2$ norm between an audio feature and an expected a priori feature [14]. A comparison between the $L_2$ norm and the maximum of the cross-correlation method of *I must be dreaming* is shown in figure 4. The cross-correlation method (right in the figure) works best when there is a tempo drift in the song, which there is in most songs.

The self-similarity matrix can now be segmented, and the segments can furthermore be clustered. In this work, the song segmentation aspect will be detailed in the following section.

## 4 Causal rhythm grouping

The grouping, or segmenting, of a song, is the task of identifying segment boundaries that usually corresponds to boundaries humans would identify. The rhythm grouping indicates that orchestration and timbre is, as far as possible, omitted in the grouping, and the causal approach indicates that it is intended for possible real-time applications. In particular, the causal approach could permit the use of the identified segments in real-time composition, for instance using Murphys Pattern Play framework [20]. Another possible use is the identification of the 1st verse (or any particular rhythmic segment) in DJ software, such as Mixxx [1].

On related work, Bartsch and Wakefield [2], used chroma-based features to identify the repeated segment that corresponds to the chorus using cross-correlation. Foote [9] used cosine distance self-similarity and radially-symmetric Gaussian kernel correlation as a novelty measure that identifies notes for small lags and segments for large time lags. Dannenberg [4] made a proof-of-concept using pitch extraction and a matrix representation and melodic similarity algorithm on *Naimi* by John Coltrane. As a final step, the segments were clustered on three different songs. Peeters [21] converts the self-similarity to lag time and performs 2D structural filtering to identify segments.

The task is to find segments that consist of audio with similar rhythmic structure. As it is a causal approach, there is no knowledge about the rhythmogram ahead of the current time.

The approach chosen is to calculate the cross-correlation self-similarity matrix at a small lag time around the current time position only, and to calculate the novelty function [9] at these time lags. As the segments in the self-similarity matrix consist of squares around the diagonal, the boundaries of the squares can be identified by correlation the diagonal with a kernel that has the same shape. Foote gives the option of using either a binary checkerboard kernel, or to create a radially-symmetric Gaussian kernel. No significant difference was found between the two kernels in this work. An example of the novelty measure, calculated using the checkerboard kernel and three different kernel sizes, for *I must be dreaming* is show in figure 5.

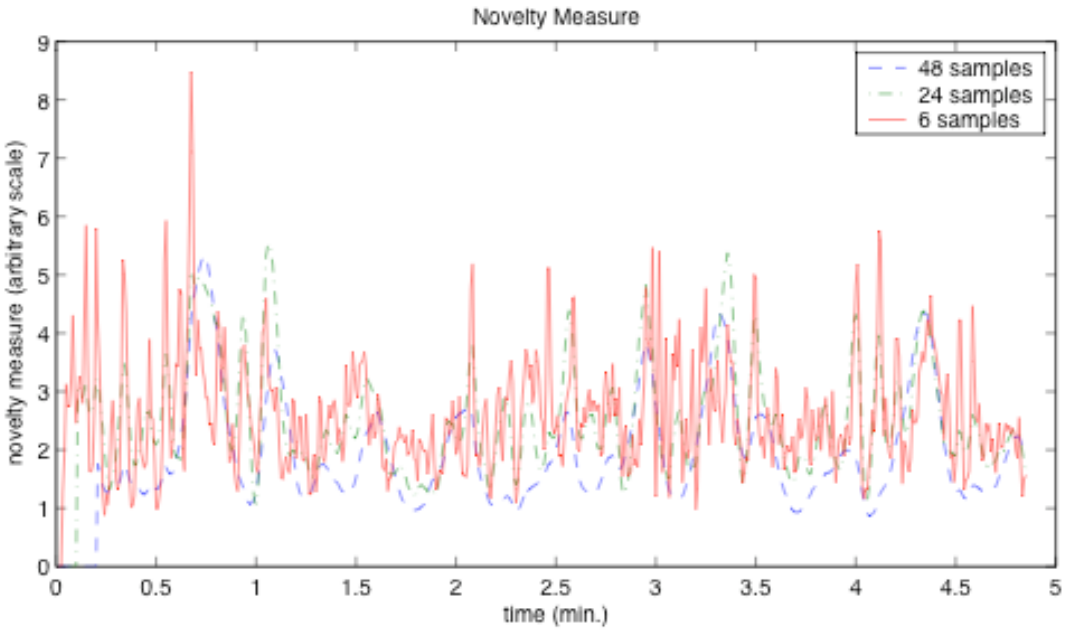


**Figure 5. Novelty Measure for *I must be dreaming* and three different checkerboard kernel sizes.**

It is clear that the small kernel sizes favors the note onsets (although only the relatively slow one, on the order of half the beat), whereas the large kernel sizes favors the structure in the song. In addition, the peaks are changing position between kernel sizes.

To identify the section boundaries, a method inspired from the scale-space community [16] in image processing is used. In this method, which, when used on images, is mimicking the way the images are blurred on a distance, the segment boundaries are found on heavily smoothed novelty measure, and the boundaries are then identified in the unsmoothed novelty measure.

The split-point time estimation is done on smoothed envelopes. The smoothing is performed by convoluting the novelty measure with a gaussian,

$$SNm_\sigma(t) = Nm * g_\sigma(t), \qquad g_\sigma(t) = \frac{1}{2\pi\sigma} e^{-\frac{t^2}{2\sigma^2}}. \tag{2}$$

The segment boundaries are now found by finding the zeros of the time derivative (with negative second derivative) of the smoothed novelty measure,

$$L_{t,\sigma}(t) == 0, L_{tt,\sigma}(t) < 0, \qquad L_{t,\sigma}(t) = \frac{\partial}{\partial t} SNm_\sigma(t), L_{tt,\sigma}(t) = \frac{\partial^2}{\partial t^2} SNm_\sigma(t). \tag{3}$$

The novelty measure is followed from the smoothed to the unsmoothed case in several steps by a method borrowed from the scale-space theory used, for edge detection, in image processing [16]. In case a peak is located near a slope, the slope influences the peak position when the novelty measure is smoothed. When the novelty function is less smoothed, it contains more noise, but the slope points correspond more to the unsmoothed case. It is thus necessary to follow the peak from the smoothed to the unsmoothed novelty measure, and to use enough smoothing steps so the slope

points can be followed. An example of the smoothing steps, and the identified segment boundaries can be seen in figure 6.

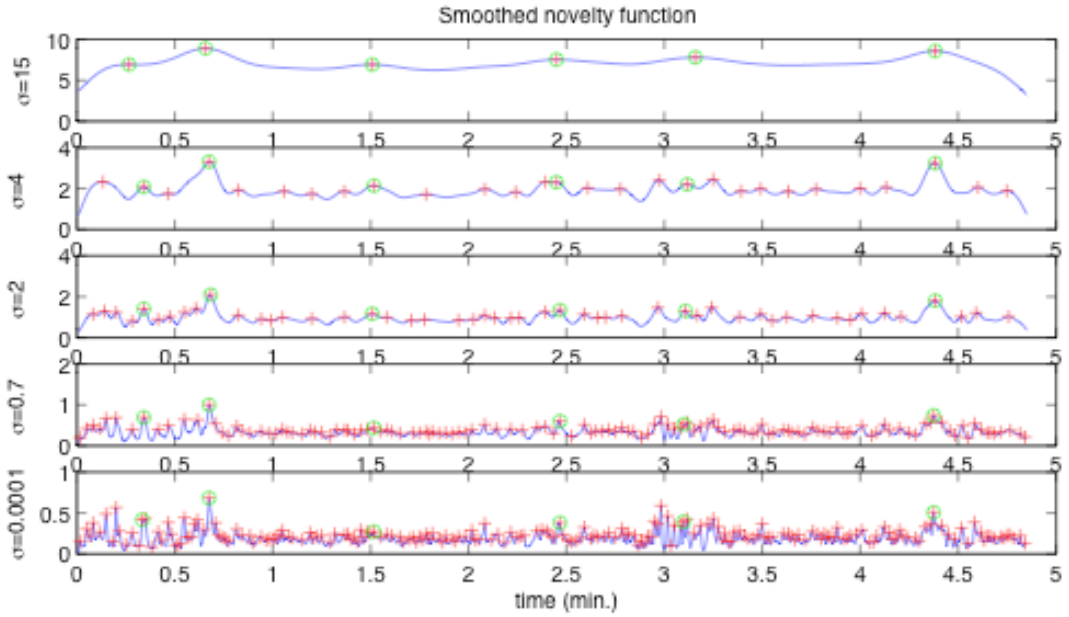


**Figure 6. Example of the smoothed novelty function, peaks ('+'), and the identified segment boundaries ('o') for *I must be dreaming*.**

Using an expert (the author) there is a certain resemblance between the intro, a segment at 3 to 3 1/2 minutes and the end segment. In addition, there are three segments consisting of verse-chorus at 0.5min to 1.5min, 1.5min to 2.5min, and 3.2min to 4.2min, the second of which the chorus lyrics is replaced with a guitar solo.

The automatic segment boundaries are found at 0, 0.2, 0.6, 1.5, 2.4, 3.3 and 4.4 minutes, where the zero and 4.4 minutes corresponds to the intro and end, the 0.6, 1.5 and 3.3 minutes corresponds to the verse chorus segments. The 0.2 minutes segment corresponds to the introduction of the vocal in the song. The second repetition of the intro theme was not found, but it seems that the automatic segmenting performs all in all almost as well as this expert. It is clear from the figure that there is much *novelty* in the song outside the found segment boundaries. More research is needed to assert whether these in fact correspond to perceptual boundaries or not. Another potential problem of the smoothing method is that it sometimes identifies a weak segment boundary in the middle of long segments, rather than a stronger boundary close to another boundary.

## 5 Evaluation

The segmentation steps are the feature extraction, the rhythmogram calculation, the self-similarity matrix calculation, the novelty measure, and the smoothing steps. The feature extraction is performed using an FFT in $O(N \log_2(N))$ steps, the rhythmogram is calculated using an autocorrelation for each 8 seconds (800 steps), which can also be performed in $O(N \log_2(N))$, the self-similarity matrix only needs to be calculated on the diagonal (4 new values for each time step), and novelty measure is smoothed in five steps. None of the last steps are very processor-intensive.

The segmentation has been performed on a small set (8) of rock and techno songs. Whereas the rock songs follow the intro, chorus, verse and break scheme well, the techno songs generally consists of long segments of music with no, or small evolutionary changes, and short consecutive segments with radical changes. The number of segments found is relatively stable for all songs, thus it seems that this method is useful for music summary, for instance. The automatic segment boundaries have been compared to human segmentation for the eight songs. First, it is obvious that some of the segment boundaries consist of vocal or other instrumental changes that are not found in the novelty measure. Around 10 % of the segment boundaries are not found, and the same amount has been misplaced by the unsmoothing peak following procedure. The smoothing makes it impossible to find short segments, which thus does not have to be prevented. Some of the misplaced peaks should possibly be found using help from some observations. For instance, it seems that some segment boundary peaks are preceded by a minima, i.e. before a change in rhythm, there is a short period with less than normal change. Another observation is that some segment boundaries are abrupt, but some consists of a gradual change where it is not clear (without counting beats and measures) where the boundary is.

The segmentation process was furthermore performed on a larger database of around three-hundred songs, consisting of child pop, pop, rock, noisy rock, world, classical, jazz, and possible other genres. A detailed analysis of the results has not been made, instead the performance of the segmentation system is evaluated using two statistics: the length of the segments, and the number of segments per song. These statistics are shown in figure 7.

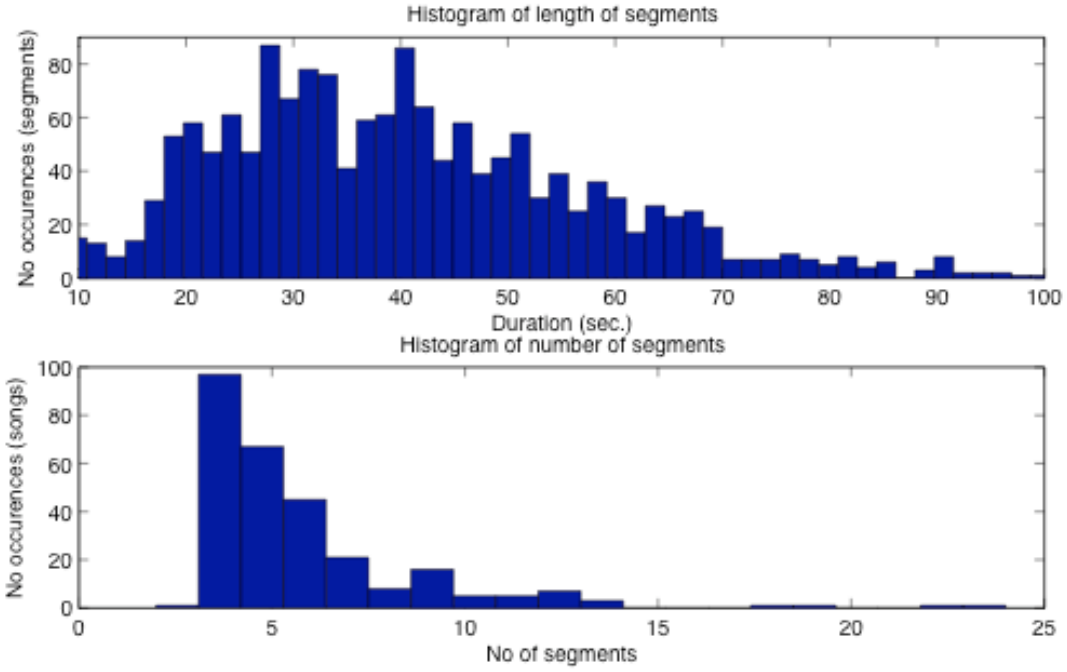


**Figure 7. Statistics of the segmentation of a large number of songs. Length of segments (top), and number of segments per song (bottom).**

It is clear that most songs are found to have a small number of segments. The extreme number of segments corresponds to four classical songs (Mozart and Schubert). No further analysis of the performance of the system in classical music has been made. An average duration of the segments of around 40 seconds seems reasonable, and although more analysis of the exact locations of the

segments boundaries is necessary, it is concluded that in most respects the system is robust and reliable.

## 6 Conclusion

This paper has presented a complete system for the estimation of segments in music. The system is based on a hierarchical model, consisting of a feature extracting step, a rhythm model, and self-similarity step and finally a segment boundary identification step. The paper introduces a feature, the Perceptual Spectral Flux (PSF) that performs twice as good as a previously used feature. The rhythmogram is an intuitive model of the rhythm that permits an instant overview of the rhythmic content of a song. It is here used as a basis for the calculation of a similarity matrix [8]. In order to minimize the processing cost for the similarity matrix calculation, an efficient segment boundary method that only uses the diagonal of the self-similarity matrix has been devised, using the novelty measure [9] and a method inspired from the scale-space community in image processing [16].

The segmentation is intended to be used in real-time recomposition, in computer-assisted DJ software, and as an automatic summary generation tool.

All the steps have been verified with formal and informal methods. The audio feature (PSF) was found to be having a signal to noise ratio twice as good as the previously used feature, the High Frequency Content (HFC). The rhythmogram was shown to illustrate the rhythm pattern throughout a song. A 2D visualization was preferred, as it enabled following of rhythm patterns that were otherwise perceived as somewhat noisy in a 3D visualization. The self-similarity using cross-correlation was preferred, as the correlation permitted a better self-similarity measure in songs with a tempo drift. Finally, the segmentation was evaluated using a small database of rhythmic songs (rock and techno). Even though some of the verse-chorus segment boundaries could not be detected, as they consist mainly of lyric differences, most of the segments were identified correctly. An added benefit of this model is that it always identifies a suitable number of segments.

## References

1. Andersen, T., H., Mixxx: Towards novel DJ interfaces, In proceedings of the New Interfaces for Musical expression, pp 30-35, 2003.
2. Bartsch, M. A. and Wakefield, G.H., To Catch a Chorus: Using Chroma-Based Representations For Audio Thumbnailing. *in* Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics (CD), 2001, IEEE.
3. Casey, M.A.; Westner, W., Separation of Mixed Audio Sources by Independent Subspace Analysis, International Computer Music Conference (ICMC), pp. 154-161, August 2000
4. Dannenberg, R., ``Listening to `Naima': An Automated Structural Analysis of Music from Recorded Audio,'' In Proceedings of the 2002 International Computer Music Conference. San Francisco, pp. 28-34, 2002.
5. Deliege, I., Melen P., Cue abstraction in the representation of musical form *in* Perception and cognition of music, edited by Irène Deliège, John Sloboda. Hove, East Sussex, England. Psychology Press, pp. 387-412, 1997.
6. Desain P., A (de)composable theory of rhythm. Music Perception, 9(4), pp 439-454, 1992.
7. Dubnov, S., Assayag, G., El-Yaniv, R., Universal Classification Applied to Musical Sequences. Proc. of the International Computer Music Conference, Ann Arbour, Michigan, 1998.

8. Eckmann, J. P., Kamphorst, S. O., and Ruelle, D., Recurrence plots of dynamical systems, Europhys. Lett. 4, 973, 1987.
9. Foote, J., Automatic Audio Segmentation using a Measure of Audio Novelty. In Proceedings of IEEE International Conference on Multimedia and Expo, vol. I, pp. 452-455, July 30, 2000.
10. Foote, J., Visualizing Music and Audio using Self-Similarity. In Proceedings of ACM Multimedia, Orlando, Florida, pp. 77-80, 1999.
11. Goto M., and Muraoka, Y., A real-time beat tracking system for audio signals. In Proceedings of the International Computer Music Conference, pp. 171-174, 1995.
12. Goto, M., and Muraoka, Y., Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. Speech Communication, Vol 27. pp. 311-335, 1998.
13. Hermansky H., Perceptual linear predictive (PLP) analysis of speech, J. Acoust. Soc. Am., vol. 87, no. 4, pp. 1738-1752, Apr. 1990.
14. Jean Laroche., Efficient tempo and beat tracking in audio recordings, J. Audio Eng. Soc., 51(4), pp. 226-233, April 2003.
15. Jensen K. , T. H. Andersen. Real-time beat estimation using feature extraction. In Proceedings of the Computer Music Modeling and Retrieval Symposium, Lecture Notes in Computer Science. Springer Verlag, pp 13-22, 2003.
16. Lindeberg, T., "Edge detection and ridge detection with automatic scale selection", CVAP Report, KTH, Stockholm, 1996.
17. Masri, P., and A. Bateman., Improved modelling of attack transient in music analysis-resynthesis. In Proceedings of the International Computer Music Conference, pages 100-104, Hong-Kong, 1996.
18. McAdams, S., Musical similarity and dynamic processing in musical context. Proceedings of the ISMA (CD), Mexico City, Mexico, 2002.
19. McAuley, J. D., Ayala, C., The effect of timbre on melody recognition by familiarity. Meeting of the A.S.A., Cancun, Mexico (abstract), 2002.
20. Murphy. D., Pattern play. In Alan Smaill, editor, Additional Proceedings of the 2nd International Conference on Music and Artificial Intelligence, On-line tech. report series of the University of Edinburgh, Division of Informatics, Edinburgh, Scotland, UK, September 2002. http://dream.dai.ed.ac.uk/group/smaill/icmai/b06.pdf.
21. Peeters, G., Deriving musical structures from signal analysis for music audio summary generation: sequence and state approach. *In* Computer Music Modeling and Retrieval (U. K. Wiil, editor). Lecture Notes in Computer Science, LNCS 2771, pp. 143-166, 2003.
22. Rauber, A., Pampalk, E., and Merkl D., Using Psycho-Acoustic Models and Self-Organizing Maps to Create a Hierarchical Structuring of Music by Musical Styles, Proceedings of the ISMIR, Paris, France. October 13-17, pp 71-80, 2002.
23. Scheirer, E., Tempo and Beat Analysis of Acoustic Musical Signals, Journal of the Acoustical Society of America, Vol. 103, No. 1, pp. 588-601, 1998.